IN THE CLAIMS

CLEAN COPY OF CLAIMS

Claims 1, 9, and 21 have been amended. Claims 6 and 18 have been cancelled. A clean copy of the claims is provided below. A marked up version of the claims follows the clean copy.

1. (Amended) A method of executing a transaction task within a transaction processing system, the method including:

responsive to an event, identifying a workflow associated with the event;

distributing a task, that at least partially executes the workflow, to an available thread within a pool of threads operating within a multiprocessor system;

identifying a processor affinity attributed to the task; and

assigning the available thread to a processor within the multiprocessor system according to the processor affinity attributed to the task.

2. (Unamended) The method of claim 1 wherein the event comprises a transaction event and the task comprises a transaction task responsive to a transaction request associated with the transaction event.

3. (Unamended) The method of claim 2 wherein the transaction task comprises a transaction routing task that routes the transaction request associated with the transaction event to an agent of the transaction processing system.

4. (Unamended) The method of claim 2 within the transaction task comprises a transaction information task to either store or retrieve information pertinent to a transaction.

5. (Unamended) The method of claim 1 wherein the task has a real-time priority and is distributed in accordance with the real-time priority to the available thread within the pool of threads.

6. (Cancelled)

7. (Unamended) The method of claim 1 including assigning the available thread to a processor within the multiprocessor system according to a thread priority.

8. (Unamended) The method of claim 7 including assigning the thread priority to the available thread based on a priority of the task distributed to the available thread.

9. (Amended) Apparatus for executing a transaction task within a transaction processing system, the apparatus comprising:

a dispatcher to identify a workflow associated with an event; and

a thread within a pool of threads operating within a multiprocessor system to execute a task that at least partially executes the workflow associated with the event,

the dispatcher to identify a processor affinity attributed to the task, and to assign the thread to a processor within the multiprocessor system according to the processor affinity attributed to the task.

10. (Unamended) The apparatus of claim 9 wherein the dispatcher generates the task that at least partially executes the workflow.

11. (Unamended) The apparatus of claim 10 including a task queue to which the task is dispatch by the dispatcher, and from which the thread within the pool of threads receives the task.

12. (Unamended) The apparatus of claim 11 including a scheduler that issues the task from the task queue to the thread within the pool of threads.

13. (Unamended) The apparatus of claim 12 wherein the scheduler issues the task from the task queue to the thread within the pool of threads based on a priority associated with the task.

14. (Unamended) The apparatus of claim 13 wherein the scheduler issues the task from the task queue according to a priority dynamically assigned to the task.

15. (Unamended) The apparatus of claim 13 wherein the scheduler issues the task from the task queue according to a real-time priority assigned to the task.

16. (Unamended) The apparatus of claim 9 wherein the task comprises a transaction routing task that routes a transaction request associated with the event to an agent of the transaction processing system.

17. (Unamended) The apparatus of claim 9 within the task comprises a transaction information task to either store or retrieve information pertinent to a transaction.

18. (Cancelled)

19. (Unamended) The apparatus of claim 9 including to assign the thread to a processor within the multiprocessor system according to a thread priority.

20. (Unamended) The apparatus of claim 19 including assigning the thread priority to the thread based on a priority of the task distributed to the thread.

21. (Amended) A method of operating a transaction processing system employing a multiprocessor architecture, the method including:

> establishing a queue of tasks, the queue of tasks including tasks for
>
> > both system and transactional functions, and
>
> servicing the queue of tasks utilizing a pool of threads executable
>
> > within a symmetric multiprocessor environment.

22. (Unamended) The method of claim 21 wherein the tasks for the system functions include any one of reporting, administration or maintenance tasks performed within the transaction processing system.

23. (Unamended) The method of claim 21 wherein the tasks for the

transactional functions include any one of routing, transaction data storage or transaction data retrieval tasks performed to facilitate a transaction within the transaction processing system.